IFW

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/796,968
Filing Date	March 11, 2004
First Named Inventor	Cheng-Yin Lee
Art Unit	2665
Examiner Name	Unknown
Attorney Docket Number	ALC 3119

Total Number of Pages in This Submission: 3

ENCLOSURES (Check all that apply)

- [ ] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [x] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) ______
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
- [ ] Proprietary Information
- [ ] Status Letter
- [x] Other Enclosure(s) (please Identify below):

Claim to Priority

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Kramer & Amado, P.C.
Signature	Terry W. Kramer
Printed name	Terry W. Kramer
Date	November 22, 2006
Reg. No.	41,541

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name		Date	

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	:	Cheng-Yin Lee
For:	:	ETHERNET PATH VERIFICATION
Serial No.	:	10/796,968
Filed	:	March 11, 2004
Art Unit	:	2665
Examiner	:	Unknown
Attorney Docket No.	:	ALC 3119
Confirmation No.	:	4035

CLAIM TO PRIORITY UNDER 37 C.F.R. 119
TRANSMITTAL OF CERTIFIED COPY OF PRIORITY DOCUMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

Applicants have claimed priority of Application No. CA 2,422,258 filed March 14, 2003 in Canada, under 35 U.S.C. § 119. In support of this claim, a certified copy of said application is submitted herewith.

No fee is believed to be due for this submission. Should any fees be required, however, please charge our deposit account number 50-0578 and please credit any excess fees to such Deposit Account

Respectfully submitted,
KRAMER & AMADO, P.C.

Date: November 22, 2006

Terry W. Kramer
Reg. No. 41,541

KRAMER & AMADO, P.C.
1725 Duke Street, Suite 240
Alexandria, VA 22314
Telephone No.: (703) 519-9801
Facsimile No: (703) 519-9802

Office de la propriété intellectuelle du Canada

Un organisme d'Industrie Canada

Canadian Intellectual Property Office

An Agency of Industry Canada

Bureau canadien des brevets

Certification

Canadian Patent Office

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,422,258**, on March 14, 2003, by **ALCATEL CANADA INC.**, assignee of Chen-Yin Lee, for "Ethernet Route Trace".

Tracy Paulhus

Agent certificateur/Certifying Officer

March 10, 2004

Date

Canada

(CIPO 68)
04-09-02

OPIC CIPO

ABSTRACT

The present invention provides an OAM tool that enables a network operator to trace the path that an Ethernet frame traverses through bridges in a bridge Ethernet LAN. Each bridge within the LAN has a control plane which examines a query message received from a previous node and determines the next node in the route to a destination. The identity of the next node is returned to the source together with a time stamp. The process is repeated until the identity of all bridges in the path has been obtained.

ETHERNET ROUTE TRACE

FIELD OF THE INVENTION

[0001] This invention relates to data communications networks and more particularly to methods and system for tracing the path an Ethernet frame traverses in a bridged Ethernet local area network (LAN).

BACKGROUND

[0002] The Ethernet system was initially developed to provide communication between a limited number of stations in a local area network (LAN) environment. As transmission medium technology and the related infrastructure have improved in recent years the speed at which Ethernet frames can be transported has dramatically increased. The distances over which Ethernet frames can be carried has also increased with improvements in system architecture.

[0003] Generally the Ethernet system consists of three basic elements: the physical medium to carry Ethernet signals between customer nodes via intermediate switches and bridges; a set of medium access control (MAC) rules embedded in each Ethernet interface; and an Ethernet frame that consists of a set of bits used to carry data including control information over the system.

[0004] Typically, each Ethernet interface, such as a bridge or switch, maintains a management information base (MIB) which stores relevant information regarding each bridge and the identity of other bridges in the system which it can access.

[0005] There are occasions in which it is desirable, or indeed essential, to be able to actually trace or track the route taken by an Ethernet frame as it travels through bridges in an Ethernet system. The capability of being able to actually trace the

path that a frame traverses is important in trouble shooting defects in the system such as those which might cause excessive packet delays or discrepancies between the MIB, control plane and data path (H/W) forwarding tables in Ethernet bridges or switches. Furthermore, an Ethernet route trace capability is important for collecting route statistics necessary for network engineering.

[0006] There is no known solution to directly trace a data path route through Ethernet bridges.

[0007] Some prior art methods find out the path a frame should traverse in a bridged LAN by querying the MIB of the bridges. However, the downside to this method is that there may be discrepancies between the MIB and the actual path a frame traverses (i.e. the data path). These discrepancies could arise because the MIB, control plane, and data path tables are out of agreement for some reason, which could actually be the cause of the problem that is being investigated. Therefore, a method that traces the exact path that a frame traverses through a bridged LAN is necessary.

[0008] Another prior art proposal is to perform an Ethernet trace route in a similar manner to an IP trace route. According to this method frames are repeatedly sent along the route, and each successive frame gets one hop closer to the destination before a bridge at that current hop responds to the sender of the frame. This method is accomplished by sending multicast frames that include a time-to-leave (TTL) variable that gets decremented at each hop. When a bridge receives a frame it decrements the TTL variable, and if the variable is expired the bridge responds to the sender. However, the problem with this approach is that the control plane (which is software driven) at each hop must process the frame, since it is not feasible to upgrade all hardware or Network Processors of bridges in a network to

perform this new function. This adds unnecessary delay and therefore any round trip measurement would be inaccurate. Furthermore, any discrepancies between the control plane tables and the data path forwarding tables would cause the resulting route trace to be different than the actual route that a frame would take along the data path, i.e. the resulting route trace would be incorrect.

[0009] Other existing partial proposals require hardware or Network Processor changes in intermediate bridges to do a trace route. While it may be feasible to update edge Network Elements, it may not be feasible to upgrade all core or intermediate bridges to do this.

[0010] The challenge is to find a way to do an accurate Ethernet trace route of a data frame path without making hardware changes to the bridges in the network.

[0011] Proposals that require hardware or Network Processor changes in intermediate bridges to do a trace route are problematic because it may not be feasible to upgrade all core or intermediate bridges to support the route trace capability.

[0012] Proposals that use the control plane to implement IP-like route traces based on TTL would not provide an accurate round trip time and could return an erroneous route, unless hardware or Network Processors are upgraded to perform these functions.

SUMMARY OF THE INVENTION

[0013] According to the present invention there is provided an OAM (operations, administration and management) tool that will enable a network operator to trace the path that a frame traverses through bridges in a bridged Ethernet LAN.

[0014] According to a first aspect of the present invention there is provided a method of tracing a data path route from a source node to a destination node through multiple intermediate nodes in a bridged Ethernet system (a bridge may also be connected by non-Ethernet media, e.g. ATM virtual circuits, MPLS Label Switched Path, IP tunnels or SDH/SONET) comprising: sending a succession of Ethernet encapsulated route query messages from the source node, each message containing a media access control (MAC) address of the destination node; receiving, at route trace enabled bridges in the system, the encapsulated route query messages; determining at a control plane of the route trace enabled bridges a MAC address of a next hop bridge on route to the destination node; returning the MAC address of the next hop bridge to source node in a response message; repeating the sequence through remaining intermediate bridges until a response message indicating that the destination node has been identified; and tabulating information in the response messages.

[0015] According to a second aspect of the present invention there is provided a system for tracing a data path route from a source node to a destination node through multiple intermediate nodes in a bridged Ethernet system comprising: means for sending a succession of Ethernet encapsulated route query messages from the source node, each message containing a media access control (MAC) address of the destination node; a control plane at route trace enabled bridges in the system to receive the encapsulated route query messages; means at a control plane of the route trace enabled bridges for determining a MAC address of a next hop bridge on route to the destination node; returning the MAC address of the next hop bridge to source node in a response message; means for repeating the sequence through remaining intermediate bridges until a response message indicating that the destination node has been identified; and means for tabulating information in the response messages.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The invention will now be described in greater detail with reference to the attached drawings wherein:

[0017] Figure 1 illustrates by way of a high level block diagram the architecture of the present invention;

[0018] Figure 2 illustrates an Ethernet frame format;

[0019] Figure 3 illustrates a first trace route query message;

[0020] Figure 4 illustrates a trace route response message;

[0021] Figure 5 illustrates a second trace route query message; and

[0022] Figure 6 is a flow diagram showing the process steps of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0023] According to the solution, an operator would initiate an Ethernet traceroute from a Provider Edge (PE) device to a destination device. In the present description the Ethernet traceroute function is known by the abreviation Etraceroute. The Etraceroute would return the MAC address (and Bridge Identification) of every Bridge on the path to the destination device and the round-trip delay at every Bridge hop on the way to the destination device.

[0024] For example an operator would enter the following command:

ETraceroute DA <SA>

[0025] Where SA is the MAC Source Address and DA is the MAC Destination Address. If SA is not specified, the source address is set to the MAC Source Address of the device where the ETraceroute is invoked.

[0026] Figure 1 is a high level block diagram of a bridge Ethernet LAN for which the Ethernet route trace method is performed. The bridged Ethernet LAN of Figure 1 includes two customer equipment nodes (CE1 and CE2) communicatively connected by Ethernet bridges PE1, P2, P3, and PE4 of a provider's network. The bridges PE1 and PE4 are at the edges of the provider's network, where bridge PE1 provides network access to CE1 and bridge PE4 provides network access to CE2. This connectivity may be also shown as follows.

CE1-----PE1-----P2-----P3-----PE4 -----CE2

[0027] In this example, it is assumed that a provider wants to trace the path of a data frame from CE1 to CE2 of a VLAN with tag value 1000. The operator would initiate an Etraceroute from a Provider Edge (PE) device to a MAC address e.g. a Customer Edge (CE) device as follows:

ETraceroute CE2_DA CE1_SA

[0028] The Etraceroute should return the MAC address (and Bridge Identification) of every Bridge on the path to the destination and the round-trip delay at every Bridge hop on the way to the destination.

[0029] In general, the Etraceroute message must be sent from a PE (PE1) that is the Source (PE1) or the next hop of the Source (CE1). At a PE (PE1), an Etraceroute Query message for a destination MAC address (e.g. CE2) is created to be sent to the next bridge hop (P2) by looking up the MAC forwarding table to the destination. The Etraceroute message contains the following fields:

-the DA, i.e. CE2_DA
-the timestamp when the message is sent out from PE1

[0030] Figure 2 shows an Ethernet frame containing the media access control header and the data field. In the following description and in particular with reference to Figures 3, 4 and 5 the Ethernet frame format has been modified to show only relevant portions as they apply to the present application. Basically the traceroute message is a payload and the normal Ethernet header is prepended to the traceroute message.

[0031] The Etraceroute message is encapsulated in an Ethernet frame (A) with the SA set to CE1_SA and the DA is set to the MAC Address of P2. The EtherType is set to VLAN and the VLAN tag value is set to 1000. The sub EtherType (EtherType of the frame belonging to VLAN 1000) is set to EtherType_Traceroute. The Ethernet frame (A) is then sent to the next hop bridge P2. Figure 3 shows Ethernet frame A.

[0032] When P2 receives the Etraceroute message, it terminates the frame and sends the frame to the control plane (or higher layer entity) handling the EtherType_Traceroute.

[0033] The control plane Traceroute entity records the time that the query was received. It then looks up the next Bridge hop to CE2_DA (this address is in the

Etraceroute query message) and creates a Etraceroute response message to PE1 with the following fields:

- the timestamp when the message is received
- the next Bridge hop to CE2_DA, i.e. P3.

[0034] Then P2 encapsulates the Etraceroute response message in an Ethernet frame (B) with the SA set to P2, the DA set to the MAC address of PE1, and the EtherType set to EtherType_Traceroute, and sends the Ethernet frame (B) to PE1. Note here it is not necessary for the Etraceroute response message to be encoded like the data frame (i.e. the VLAN tag is not required). Ethernet frame B is shown in Figure 4.

[0035] When PE1 receives the Etraceroute response message, it terminates the message (since it is destined to it) and sends the message to the control plane handling the EtherType_Traceroute. At PE1, another Etraceroute Query message is created for sending the next bridge hop P3 (obtained in the Etraceroute Response message from P2). The Etraceroute Query message contains the following fields:

- the DA, i.e. CE2_DA
- the timestamp when the message is sent out from PE1

[0036] The Etraceroute Query message is encapsulated in an Ethernet frame (C) with the SA set to CE1_SA and the DA set to the MAC Address of P3, the EtherType is set to VLAN and the VLAN tag value is set to 1000. The sub EtherType (EtherType of the frame belonging to VLAN 1000) is set to EtherType_Traceroute. The Ethernet frame (C) is then sent to the next bridge hop P3. Ethernet frame C is shown in Figure 5.

[0037] When P3 receives the Etraceroute Query message, it terminates the frame and processes the EtherType_Traceroute frame and the whole procedure is repeated as shown above for each Bridge hop towards CE2, until a Etraceroute Response message is received from PE4. The Etraceroute Response message from PE4 contains the following :

- the timestamp when the message is received
- the next Bridge hop to CE2_DA, i.e. NULL.

[0038] When PE1 receives a NULL next bridge hop, the EtherType_Traceroute entity displays all the collected information as shown below.

ETraceroute displays all the Bridges as follows:

[0039] ETraceroute from PE1 to PE4 for test packet from CE1 to CE2:

PE1 to P2 : rtt - 10ms
PE1 to P3 : rtt - 15 ms
PE1 to P4 : rtt - 30 ms
PE1 to PE5: rtt - 40 ms

[0040] Figure 6 is a flow diagram showing process steps according to the invention.

[0041] A key aspect of the present invention is that the consecutive Etraceroute query messages are sent to the next hop and the subsequent next hop in the same path as a data frame. All bridges ideally should be configured to not discard or punt unknown/new EtherType such as EtherType_Traceroute to the control plane, to prevent intermediate bridges from intercepting EtherType_Traceroute messages.

[0042] It should be noted that if the SA is not specified, the Ethernet header SA is set to the PE's MAC SA.

[0043] In this solution, no hardware or Network Processor changes in bridges are required. Each bridge only need to be loaded with new application software which handles the EtherType Traceroute.

[0044] In a further aspect of the invention, if one of the bridges in the path does not have the route trace functionality the following steps are used to skip over that bridge and continue the trace. The traceroute software at ingress (source node CE1 or immediate next hop node PE1) would time out when it doesn't receive a response from a downstream bridge, and report the trace learned so far (i.e. it can't trace all the way to the destination MAC address).

[0045] The ingress bridge may issue another traceroute with the option to multicast to downstream bridges. This traceroute multicasts (a multicast address is reserved for this purpose) a query message to all downstream bridges (on the port towards the destination MAC address) and hence should be used sparingly. Etraceroute enabled bridges are members of this reserved multicast address. An intermediate bridge would receive and process the multicast query message as well as forward the multicast message. If a bridge does not understand the query message it will ignore it (but the query message is forwarded to the other downstream bridges of the spanning tree). All downstream bridges with forwarding address of the target destination MAC address should respond with the next hop bridge MAC address.

For e.g.

CE1-----PE1----P2-----P3-----P4 ----P5----PE6---CE2

[0046] If P2 and P3 are not Etraceroute enabled, P4, P5 and PE6 respond with the appropriate next hops in response to the multicast traceroute query message from PE1.

[0047] PE1 concludes this is the set of consecutive downstream bridges that it can trace towards the destination CE2, starting from P4, since each response message has a next hop which matches the MAC source address of another response message, with the exception of the egress bridge PE6, with a next hop of the destination node. PE1 then displays the bridges that it can trace, starting from P4

PE1 to unknown hop(s)
PE1 to P4 (first Etraceroute aware bridge) : rtt - 30 ms
PE1 to P5: rtt - 40 ms
PE1 to PE6 : rtt – 60 ms

[0048] If P2 , P3 and P5 are not Etraceroute enabled, P4 and PE6 respond with the appropriate next hops in response to the multicast traceroute query message from PE1.

[0049] PE1 concludes that there is a number of downstream bridges that it can trace towards the destination CE2. PE1 then displays the bridges that it can trace, starting from P4, and any other intermediate bridges that respond to the traceroute query message.

PE1 to unknown hop(s)
PE1 to P4 (first Etraceroute aware bridge) : rtt - 30 ms
PE1 to unknown hop(s)
PE1 to PE6 : rtt – 60 ms

[0050] To improve the accuracy of the traceroute, the PE1 may send (unicast) a traceroute query message to all Etraceroute bridges as described before, instead of displaying the bridge hops directly after receiving traceroute response message. The extra step ensures the traceroute message traverse the paths as a normal data packet would. The ingress would not send a traceroute query message to downstream bridges that have not responded to the multicast query message. These bridges would be skipped in the traceroute query and the traceroute software at ingress would report no responses from these bridges.

[0051] In the present invention the traceroute message is forwarded like a data frame, hence the traceroute correctly and accurately verifies the path and functional elements that are forwarding data frames.

[0052] Further, no hardware or Network Processor changes in bridges are required. Bridges are loaded with new application software that handles the EtherType Traceroute. The solution works even if some bridges in the route being traced do not have the trace route software installed.

[0053] In the unlikely event that data path changes occur during the route tracing procedure, the procedure could be run again, or several more times, in such cases. In fact, multiple tracing for the same route could be a standard option to further increase confidence in its results.

[0054] Although specific embodiments of the invention have been described and illustrated it will be apparent to one skilled in the art that numerous changes can be made thereto without departing from the basic concept. It is to be understood, however, that such changes will fall within the full scope of the invention as defined by the appended claims.

I Claim:

1. A method of tracing a data path route from a source node to a destination node through multiple intermediate nodes in a bridged Ethernet system comprising:

sending a succession of Ethernet encapsulated route query messages from the source node, each message containing a media access control (MAC) address of the destination node;

receiving, at route trace enabled bridges in the system, the encapsulated route query messages;

determining at a control plane of the route trace enabled bridges a MAC address of a next hop bridge on route to the destination node;

returning the MAC address of the next hop bridge to source node in a response message;

repeating the sequence through remaining intermediate bridges until a response message indicating that the destination node has been identified; and

tabulating information in the response messages.

2. The method as defined in claim 1 wherein when the encapsulated route query messages are received at a non-enabled route trace bridge steps are taken to skip to a route trace enabled bridge.

3. The method as defined in claim 2 wherein the service node sends a multi cast message to nodes downstream of the non-enabled bridge to locate a route trace enable bridge in the route to the destination node.

4. The method as defined in claim 3 wherein the encapsulated route query message is sent to the bridge next to the non-enabled bridge which responds to the multi cast message.

5. The method as defined in claim 1 wherein the query message includes address information of the source and destination nodes at connection type.

6. The method as defined in claim 5 wherein the query message also includes a time stamp value entered by the control plane at respective route trace enabled bridges.

7. The method as defined in claim 1 wherein the response message includes address information of the source nodes and destination node.

8. The method as defined in claim 1 wherein the step of tabulating information generates a report defining bridges traversed by the Ethernet frame.

9. The method as defined in claim 8 wherein time stamp information respecting each bridge traversed included in the report.

10. A system for tracing a data path route from a source node to a destination node through multiple intermediate nodes in a bridged Ethernet system comprising:

means for sending a succession of Ethernet encapsulated route query messages from the source node, each message containing a media access control (MAC) address of the destination node;

a control plane at route trace enabled bridges in the system to receive the encapsulated route query messages;

means at a control plane of the route trace enabled bridges for determining a MAC address of a next hop bridge on route to the destination node;

returning the MAC address of the next hop bridge to source node in a response message;

means for repeating the sequence through remaining intermediate bridges until a response message indicating that the destination node has been identified; and

means for tabulating information in the response messages.

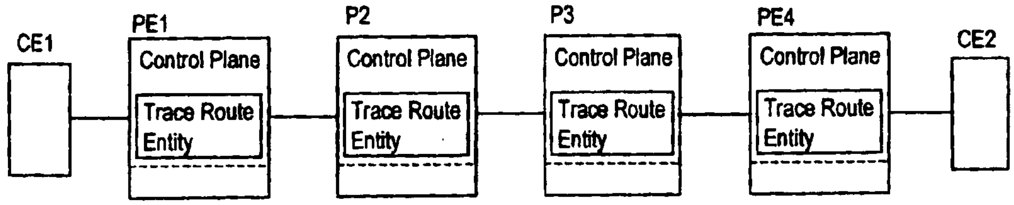

FIGURE 1

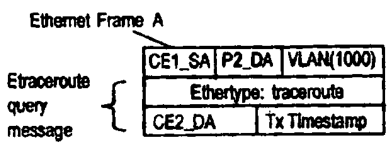

FIGURE 3

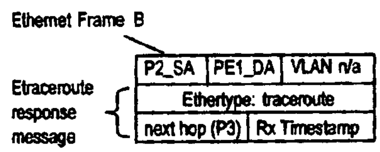

FIGURE 4

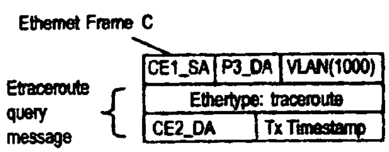

FIGURE 5

Marks & Clerk

Preamble	D addr	S addr	Type	D addr	S addr	Data	CRC
8 bytes	6 bytes	6 bytes	2 by	6 bytes	6 bytes	maximum of 1500 bytes	4 bytes

MAC

Media Access Control Header

Data Field (46-1500 bytes)

FIGURE 2

Marks & Clerk

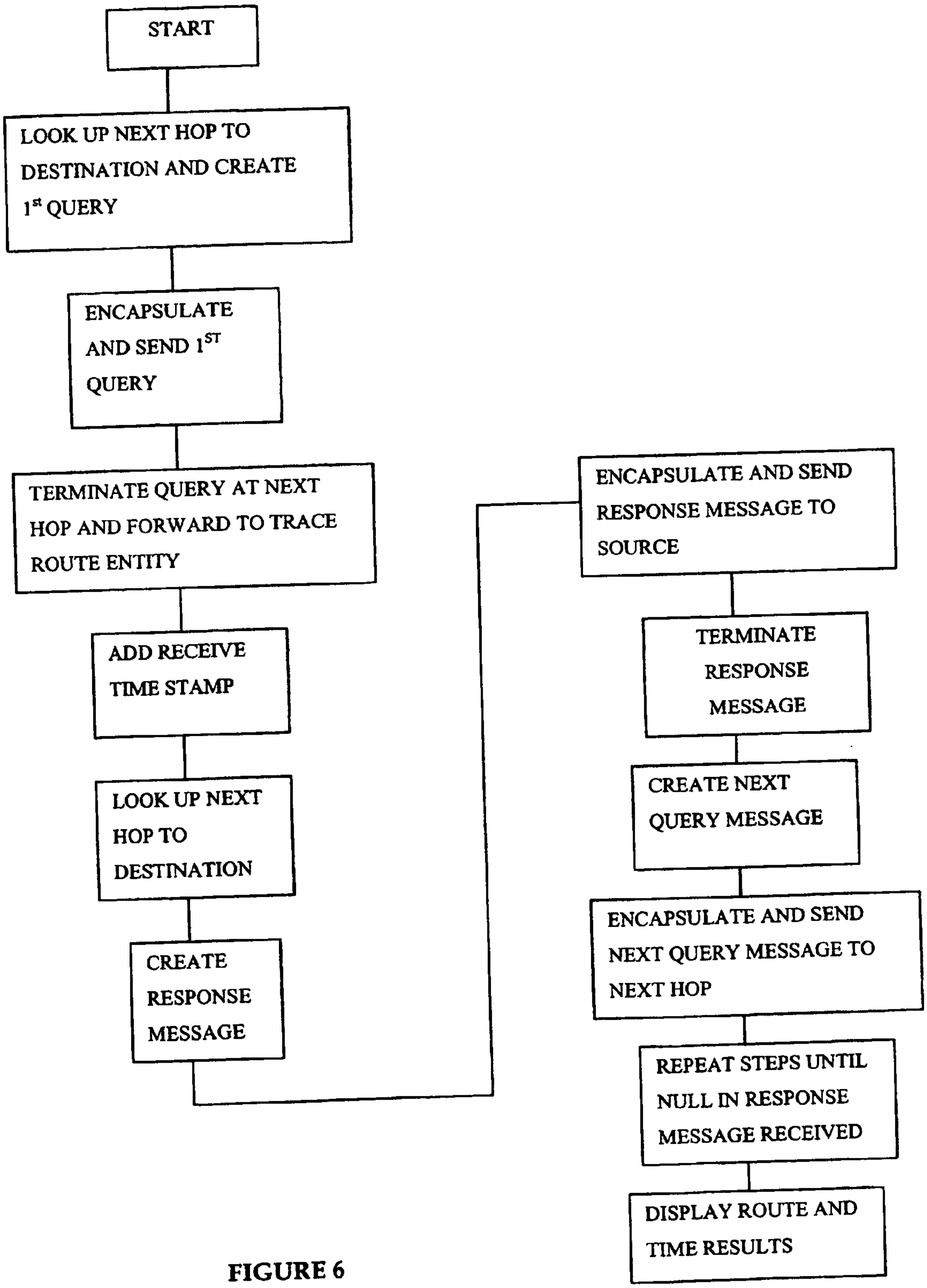

FIGURE 6

Marks & Clerk